

1. (Currently Amended) A system for loading software applications, comprising:
 - a server, executing a Java virtual machine that includes a system classloader, for storing and running a plurality of software applications, wherein each of said plurality of software applications includes a plurality of deployable modules and classes associated therewith, and wherein the software applications or modules therein can be customized by a software developer and then deployed to run on the same server;
 - a control file, that can be edited by the software developer and associated with said plurality of software applications or modules, wherein said control file specifies a hierarchy of application classloaders as children of the system classloader, to be used with the modules in said plurality of software applications, and wherein the hierarchy includes a plurality of nested branches and siblings, including providing each of said plurality of software applications or modules with its own application classloader hierarchy so that the software applications or modules are not aware of classloaders or classes that are assigned to another software application, and wherein the hierarchy is specified by the software developer to provide namespace separation between two or more of the plurality of software applications or between different modules in any one of the software applications; and
 - a deployment utility that, upon receiving a request to deploy and run a software application on the server,
 - parses the control file and determines which classloaders are specified therein for the software application being deployed,
 - loads with said software application into the Java virtual machine at the server a selection of said application classloaders corresponding to the hierarchy specified by said control file, including creating sibling classloaders for implementations whose classes are to be loaded separately as specified by the control file, and, if a particular software application or a module in a software application is being redeployed then loading only the application classloaders that are specified in the branches for that particular software application or module, without loading any of the other branches in the hierarchy, and
 - enables the server to host multiple isolated software applications or modules within the Java virtual machine, as defined by the hierarchy;
 - wherein, upon receiving a call to instantiate an implementation, the system searches within the hierarchy including first searching the calling classloader, and then successively searching parent classloaders above the calling classloader, excluding any sibling classloaders, to locate and instantiate necessary classes or objects.

2. (Previously Presented) The system of claim 1 wherein said control file can be modified by a software developer to specify a particular hierarchy of application classloaders to be used with a particular software application.
3. (Original) The system of claim 2 wherein said control file is a deployment descriptor.
4. (Original) The system of claim 3 wherein said control file is interpreted by an application container constructor during deployment so as to define the application container.
5. (Canceled).
6. (Original) The system of claim 1 wherein said hierarchy is specified by a classloader structure declaration.
7. (Previously Presented) The system of claim 1 wherein a combination of said modules may be associated with a plurality of subordinate application classloaders.
8. (Canceled).
9. (Previously Presented) The system of claim 1 wherein the server provides multiple software applications, each with their own hierarchy of application classloaders.
10. (Canceled).
11. (Currently Amended) A method for loading software applications on a server executing a Java virtual machine that includes a system classloader, comprising the steps of:
 - providing a server executing a Java virtual machine for storing and running a plurality of software applications;
 - providing a plurality of software applications, wherein each of said plurality of software applications includes a plurality of deployable modules and classes associated therewith, and wherein the software applications or modules therein can be customized by a software developer and then deployed to run on the same server;
 - providing a control file associated with said software application or modules, wherein said control file can be edited by a software developer and specifies a hierarchy of application

classloaders as children of the system classloader, to be used with the modules in said software application, and wherein the hierarchy includes a plurality of nested branches and siblings, including providing each of said plurality of software applications or modules with its own application classloader hierarchy so that the software applications or modules are not aware of classloaders or classes that are assigned to another software application, and wherein the hierarchy is specified by the software developer to provide namespace separation between two or more of the plurality of software applications or between different modules in any one of the software applications, and wherein said application classloaders are children of system classloader; and

upon receiving a request to deploy and run a software application on the server,

parsing the control file and determining which classloaders are specified therein for the software application being deployed,

retrieving a selection of said application classloaders according to the hierarchy specified by said control file, and

loading said modules and classes into the Java virtual machine at the server as part of said software application corresponding to said hierarchy, including creating sibling classloaders for implementations whose classes are to be loaded separately as specified by the control file, and, if a particular software application or a module in a software application is being redeployed then loading only the application classloaders that are specified in the branches for that particular software application or module, without loading any of the other branches in the hierarchy, to enable the server to host multiple isolated software applications or modules within the Java virtual machine, as defined by the hierarchy;

wherein, upon receiving a call to instantiate an implementation, the system searches within the hierarchy including first searching the calling classloader, and then successively searching parent classloaders above the calling classloader, excluding any sibling classloaders, to locate and instantiate necessary classes or objects.

12. (Previously Presented) The method of claim 11 wherein said control file can be modified by a software developer to specify a particular hierarchy of application classloaders to be used with a particular software application.

13. (Original) The method of claim 12 wherein said control file is a deployment descriptor.

14. (Original) The method of claim 13 wherein said control file is interpreted by an application

container constructor during deployment so as to define the application container.

15. (Canceled).

16. (Original) The method of claim 11 wherein said hierarchy is specified by a classloader structure declaration.

17. (Previously Presented) The method of claim 11 wherein a combination of said modules may be associated with a plurality of subordinate application classloaders.

18. (Canceled).

19. (Previously Presented) The method of claim 11 wherein the server provides multiple software applications, each with their own hierarchy of application classloaders.

20-30. (Canceled).

31. (Previously Presented) The system of claim 1 wherein each of the plurality of modules is one of an EJB or Web application file, together with associated classes, configuration rules and resource files for that EJB or Web application file.

32. (Previously Presented) The system of claim 1 wherein the hierarchy that includes a plurality of branches specified by the software developer further comprises a plurality of nested references to modules and/or individual class files as specified by the software developer.

33. (Previously Presented) The method of claim 11 wherein each of the plurality of modules is one of an EJB or Web application file, together with associated classes, configuration rules and resource files for that EJB or Web application file.

34. (Previously Presented) The method of claim 11 wherein the hierarchy that includes a plurality of branches specified by the software developer further comprises a plurality of nested references to modules and/or individual class files as specified by the software developer.

35-42. (Canceled).

43. (Previously Presented) The system of Claim 1, wherein the control file includes a tag layout, and application class-loader declaration, including structure elements, that determine the hierarchy of modules and classes of the software application to be loaded into the application server; and

wherein upon deployment, the deployment utility retrieves modules and classes from a computer readable medium in a manner consistent with the tag layout in the configuration file, and constructs an application container at the server with the classes and modules, in the order in which the classes and modules were retrieved, and according to the specified hierarchy, to create a hierarchical classloader.

44. (Previously Presented) The method of Claim 11, wherein the control file includes a tag layout, and application class-loader declaration, including structure elements, that determine the hierarchy of modules and classes of the software application to be loaded into the application server; and

wherein upon deployment, the deployment utility retrieves modules and classes from a computer readable medium in a manner consistent with the tag layout in the configuration file, and constructs an application container at the server with the classes and modules, in the order in which the classes and modules were retrieved, and according to the specified hierarchy, to create a hierarchical classloader.

45. (Currently Amended) A computer readable storage medium, including instructions stored thereon which when read and executed by a computer cause the computer to perform the steps comprising:

providing a server executing a Java virtual machine for storing and running a plurality of software applications;

providing a plurality of software applications, wherein each of said plurality of software applications includes a plurality of deployable modules and classes associated therewith, and wherein the software applications or modules therein can be customized by a software developer and then deployed to run on the same server;

providing a control file associated with said software application or modules, wherein said control file can be edited by a software developer and specifies a hierarchy of application classloaders as children of the system classloader, to be used with the modules in said software application, and wherein the hierarchy includes a plurality of nested branches and siblings, including providing each of said plurality of software applications or modules with its own application

classloader hierarchy so that the software applications or modules are not aware of classloaders or classes that are assigned to another software application, and wherein the hierarchy is specified by the software developer to provide namespace separation between two or more of the plurality of software applications or between different modules in any one of the software applications, and wherein said application classloaders are children of system classloader; and

upon receiving a request to deploy and run a software application on the server,
parsing the control file and determining which classloaders are specified therein for the software application being deployed,
retrieving a selection of said application classloaders according to the hierarchy specified by said control file, and

loading said modules and classes into the Java virtual machine at the server as part of said software application corresponding to said hierarchy, including creating sibling classloaders for implementations whose classes are to be loaded separately as specified by the control file, and, if a particular software application or a module in a software application is being redeployed then loading only the application classloaders that are specified in the branches for that particular software application or module, without loading any of the other branches in the hierarchy, to enable the server to host multiple isolated software applications or modules within the Java virtual machine, as defined by the hierarchy;

wherein, upon receiving a call to instantiate an implementation, the system searches within the hierarchy including first searching the calling classloader, and then successively searching parent classloaders above the calling classloader, excluding any sibling classloaders, to locate and instantiate necessary classes or objects.

46. (Previously Presented) The computer readable storage medium of Claim 45, wherein the control file includes a tag layout, and application class-loader declaration, including structure elements, that determine the hierarchy of modules and classes of the software application to be loaded into the application server; and

wherein upon deployment, the deployment utility retrieves modules and classes from a computer readable medium in a manner consistent with the tag layout in the configuration file, and constructs an application container at the server with the classes and modules, in the order in which the classes and modules were retrieved, and according to the specified hierarchy, to create a hierarchical classloader.